AHLF: ad hoc learning of fragmentation patterns in mass spectra detects post-translationally modified peptides prior to database search

Tom Altenburg [1,2,3], Shengbo Wang [1,2,4], Thilo Muth [1,5] and Bernhard Y. Renard [1,3]*

[1]Bioinformatics Unit (MF1), Robert Koch Institute, Berlin, Germany
[2]Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[3]Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Potsdam, Germany
[4]Department of Computer Science, Vrije University Amsterdam, The Netherlands
[5]Federal Institute for Materials Research and Testing (BAM), Berlin, Germany

*To whom correspondence should be addressed.

Abstract

Motivation: Publicly available mass spectrometry-based proteomics data has grown exponentially in the recent past. Yet, large scale spectrum-centered analysis usually involves predefined fragmentation features that are limited and prone to be biased. Using deep learning, the decision making for a suitable fragmentation model can be carried out in a data-driven manner.
Results: We introduce a framework that allows end-to-end training of generic deep learning models on a large collection of high resolution tandem mass spectra. In this case we used 19.2 million labeled spectra from more than a hundred individual PRIDE repositories. In our framework, we developed a representation that captures the complete information of a high-resolution spectrum facilitating a loss-less reduction of the number of features largely independent of the actual resolution. Additionally, it allows us to use common trainable layers, e.g. recurrent or convolutional operations. Specifically, we use a deep network of stacked dilated convolutions to model long range associations between any peaks within a tandem mass spectrum. We exemplify our approach by learning to detect post-translational modifications - in this case, protein phosphorylation - only based on a given mass spectrum in a fully data-driven manner. To the best of our knowledge, this is the first end-to-end trained deep learning model on tandem spectra that is able to ad hoc learn fragmentation patterns in high-resolution spectra. Our approach outperforms the current state-of-the-art in predicting if a mass spectrum originates from a phosphorylated peptide.
Availability: Our deep learning framework is implemented in tensorflow. The open source code including trained weights is available at gitlab.com/dacs-hpi/ahlf
Contact: bernhard.renard@hpi.de

1 Introduction

Publicly available mass spectrometry (MS)-based proteomics data has grown exponentially in terms of number of data sets and amount of data Vizcaíno *et al.* (2017). Still, the application of deep learning models directly to fragmentation spectra has not yet sparked in the community. Most high-throughput proteomics studies primarily use LC-MS/MS to identify peptides from fragmentation spectra (also ms/ms or tandem spectra), followed by a conventional database search to perform peptide identification Aebersold and Mann (2016). In a database search, an acquired mass spectrum is scored against a list of in-silico digested

© The Author 2020.

candidate peptides. For each peptide candidate, a theoretical spectrum is constructed and compared to the acquired experimental spectrum Nesvizhskii *et al.* (2007). The list of candidate peptides (i.e. the so-called search space) increases when post-translational modifications (PTMs) are taken into account. The search space is exponentially inflated when multiple modification sites are considered. Various strategies exist in order to reduce the search space because the peptide-spectrum matching process is the most expensive step in a database search Kong *et al.* (2017); Bittremieux *et al.* (2018); May *et al.* (2018); Devabhaktuni *et al.* (2019).

In contrast, only a few tools make predictions purely based on the spectrum itself and are therefore independent of a database Lu *et al.* (2008); Dorl *et al.* (2018). In particular, calling the presence of post-translational modifications can be separated from the sequence retrieval. A collection of features and patterns exist independently of the actual peptide sequence or in an equi-variant manner (e.g. shifts appear in-line with the position of a PTM within a sequence - but the presence of a shift is independent of the sequence) Zolg *et al.* (2018); Potel *et al.* (2019). Those patterns give rise to whether the peptide carries a specific PTM or not. Therefore, calling a PTM based on a given spectrum is a binary classification problem.

To the best of our knowledge, there have not been attempts to directly present tandem mass spectra to a deep learning model and ad hoc learning essential fragmentation patterns due to physico-chemical properties of the peptide at hand. From our point of view, the notion and the success of deep learning is based on the following principles: (i) feature creation and extraction is integrated into the learning process (commonly called 'end-to-end'-learning) and (ii) the performance scales with amount of training data. In reality, this is only met when the architecture has a build-in trainable redundancy that matches the nature of the problem. Also, the learning process including the used tensor operations have to be efficient enough to cope with the vast amount of training data LeCun *et al.* (2015); Xu *et al.* (2020). Yet, for complex and high-dimensional data points this is mostly ever successful when parameter sharing techniques such as convolutional or recurrent neural networks are used. As those are efficiently implemented in up-to-date deep learning libraries. Regarding biological data, deep learning approaches on imaging and sequential data are very successful, in the sense that work on those types of data is published in a high frequency. We argue this is mainly due to the straight-forward applicability of findings from computer vision and natural language processing to imaging and sequential data. Interestingly, there are at least two applications of deep learning models that are well-received by the proteomics community: (i) fragment intensity prediction and (ii) de novo sequencing. Yet, the models that are used for both applications are in their essence sequence models. A recurrent neural network applied to a peptide sequence predicts fragment intensities as implemented in pDeep Zhou *et al.* (2017) and Prosit Gessulat *et al.* (2019). Similarly, a recurrent neural network facilitates de novo sequencing by guiding a dynamic programming approach with a learned heuristic of peptide sequence patterns in case of DeepNovo Tran *et al.* (2017). The latter also relies on a convolutional network that extracts features from the mass spectrum, but it is unable to ad hoc learn fragmentation, e.g. the various ion-types of amino acids are built-in and not trainable parameters.

A generic end-to-end training of deep learning models directly applied to tandem mass spectra could greatly benefit the proteomics community Xu *et al.* (2020). Here, we developed (i) a loss-less spectrum representation that allows efficient model training and (ii) set up a capable model architecture that accounts for the high resolution and complex nature of tandem mass spectra. Our method relies on tensor operations that are commonly implemented in most deep learning libraries.

We exemplify a specific use-case by training a PTM-caller on an extensive data set of (un-)phosphorylated peptides. However, it should be noted that we do not make any specific assumptions about phosphorylations or PTMs in general. Instead, we set up a generic architecture that is by design unspecific at first and then becomes application-specific - in this case, classifying spectra as phosphorylated. We argue that a comprehensive but more generic spectrum model has the following advantages: (i) it is not limited to a fixed set of ion types, (ii) it does not exclude internal fragments and neutral losses, (iii) it is possible to learn non-proteinogenic amino acids Cvetesic *et al.* (2016) and to adjust the alphabet size accordingly Kreitzberg *et al.* (2019), (iv) it allows unrestricted study of post-translational modifications and (v) it is orthogonal to a database search (e.g. can be used to reduce search space).

Our approach exceeds the performance of the current state-of-the-art in predicting if a tandem spectra originates from a phosphorylated peptide. Currently, spectrum-centered methods (not incorporating a database search) are based on engineered PTM-specific features from MS/MS spectra and use traditional machine learning including support vector machine and random forest models to distinguish between modified and unmodified peptides Lu *et al.* (2008); Dorl *et al.* (2018). In contrast, our method is enabled by the use of deep learning. To our best knowledge, this makes our approach the first end-to-end trained deep learning model on tandem spectra – being able to ad hoc learn fragmentation patterns in high-resolution spectra. Similarly, our approach can easily be adapted to other applications and may further inspire upcoming end-to-end training on tandem mass spectra. An implementation with command-line instructions is publicly available from gitlab.com/dacs-hpi/ahlf and includes four independently trained models (e.g. can be used as an ensemble to further boost performance). Additionally, the code-repository allows fine-tuning of AHLF as well as training from scratch on third-party or user-specific data.

2 Methods

2.1 Spectrum representation

Here, we describe the rationale of our spectrum representation. Mass spectrometers are highly accurate in performing m/z measurements. For example, a relative error of 20 ppm (this error is instrument specific) means that two peaks at around 500 m/z can be separated if they differ in at least 0.01 Da Strupat *et al.* (2013); Wolski *et al.* (2005). Modern instruments are also able to reach sub-ppm mass accuracy Wenger *et al.* (2010). A naive conversion of a peaks-list, which is a sparse vector, into a dense vector would result in a vector with multiple hundred-thousand entries (e.g. window-size=2,000 and smallest difference=0.01 Da results in a vector with 200,000 entries) of which most entries are zero. Still, we are interested in such a representation as the location of a feature reflects the m/z-values. It is generally easier to find useful decision boundaries, especially in a deep learning framework. With a spatial distribution of features, a neuron only has to distinguish between a feature being present or not being present.

Additionally, the spectrum representation should reflect a regular grid, which is equally spaced and with fixed connectivity of m/z-values. Such a representation is comparable to an image or a time-series with constant time-steps. This makes a spectrum compatible with a network that uses convolutional or recurrent layers. Otherwise, using a peaks-list as-is would require, for instance, graph-convolutions which are less well optimized on the hardware-level. Also a graph representation would obscure m/z-values as continuous features.

Also, we are seeking a loss-less representation, i.e. not loosing peaks and not obscuring resolution in m/z. Below we study and quantify the potential loss of peaks by comparing the original centroided spectrum (a peaks-list) to our representation in a one-to-itself setting. In contrast, we do not perform our comparison in a one-to-one (pairwise) or even one-to-many setting. The one-to-one setting is commonly used, as almost any

database search performs pairwise comparison of an acquired spectrum against a given theoretical spectrum Wolski *et al.* (2005). The one-to-one and one-to-many setting compare more than one spectrum and therefore highlight potential conflicts between peaks from various spectra. The one-to-itself setting is not able to quantify such conflicts, e.g. between co-localized peaks across an entire data set.

A centroided tandem spectrum is a list of peaks that are tuples of mass-over-charge (m/z) and intensity (I). This does not fulfill the above stated requirements (distributing peaks spatially, being a fixed grid and being loss-less). Hence, for our representation we convert a each peaks-list by (i) populating a dense feature vector according to the peaks-list, imputing any empty entry with zeros and (ii) applying the arg-maximum and maximum within small mass segments. This yields two feature vectors, one containing intensities and the other the m/z-remainder, the exact position within the respective segment. In fact, if the sparsity matches the segment size (meaning exactly one peak per segment) this operation is reversible as illustrated in Fig. 1A. In order words, one is able to recover the original peaks-list from our two-vector representation. To achieve a truly loss-less conversion we can exploit the sparsity of centroided spectra and adjust the segment size accordingly. Theoretically, the chance of two peaks randomly falling into the same segment is marginal. A peaks-list contains l, total number of peaks. Typically, a ms/ms spectrum has hundreds, in extreme cases up to a few thousands of centroided peaks l and a dense vector matching the instrument resolution has easily multiple hundreds-of-thousands of valid m/z-bins, denoted by b. Assuming a uniform population of the dense m/z-vector, the probability for randomly choosing a m/z-bin that has been occupied already is given by $1/(b-l)$. It is known that the m/z dimension displays dead spots (e.g. combined histogram of m/z from all peaks-lists). Hence, m/z is usually not uniformly populated in a global manner. However, looking at a single tandem spectrum we can see that peaks are not clumped together. Extreme cases like poorly fragmented spectra are usually discarded during quality assessment as those are not very informative. In fact, for more than 1.2 million spectra studied in Fig. 1B, not a single spectrum lost a peak when we use segments smaller than 0.6 Da. For this survey in Fig. 1B we included all spectra (also including spectra before database search based filtering) from the Lung data set of PXD012174.

With the segment-size one can control two important aspects of the two-vector spectrum representation: (i) the risk to lose peaks, (ii) the total number of features. The trade-off between these two effects is shown in Fig. 1B. Based on this data, we selected a segment-size of 0.5 Da resulting in a small number of features while not losing a single peak. Additionally, we chose a fixed window-size of 100-1900 m/z yielding 7200 features in total. Hence, all spectra are represented as a tensors of size (3600,2).

To handle the amount of spectra, especially for feeding a GPU with training samples efficiently, we set up a custom pipeline facilitating the conversion of peaks-lists into the two-vector representation. It also performs common preprocessing steps, e.g. ion-current normalization that is dividing intensities by the sum of squared intensities. Our preprocessing pipeline is implemented in Python, largely integrating Pyteomics Goloborodko *et al.* (2013); Levitsky *et al.* (2019) and tensorflow. The combined preprocessing and training pipeline can be found as part of our code repository. Our pipeline accepts mgf-files as input-files. Spectra from raw-files were centroided and converted to mgf-files by using ThermoRawFileParser Hulstaert *et al.* (2020).

2.2 Model architecture

Fragmentation patterns in tandem mass spectra are predominantly long-range associations between pairs of peaks or multiple peaks. For example, the fragmentation of a phosphopeptide often involves the loss of phosphoric acid H_3PO_4, which results in a delta mass of 98 Da between pairs of ions Degnore and Qin (1998); Potel *et al.* (2019). We consider those long-range relations as those peak-pairs are separated by multiple ten or hundred Dalton and hence are far apart in the feature vector in Fig. 1C. Additionally, a mass spectrum can have multiple peak-pairs with the same delta mass that are present at different locations in the mass spectrum. The location is equi-variant with the position of a phosphorylation in the sequence. In contrast, diagnostic ions (also low-mass or reporter ions) have defined locations in the lower mass region (below 300 Da). Those are usually immonium ions, comprised of a single side chain. e.g. the immonium ion of phosphoserine can be observed at 140 Da Falick *et al.* (1993). For some PTMs the molecule of the modification itself can further fragment, which results in characteristic patterns for the different types of modifications Zolg *et al.* (2018).

Traditional methods, including de novo sequencing or database search, commonly only consider a limited set of sequencing ions and a limited set of ion-types. Similarly, for phosphopeptide prediction, Colander uses 5 features, whereas PhoStar uses 60 features Lu *et al.* (2008); Dorl *et al.* (2018). These features are derived from either immonium ions or pair-wise mass differences between ions. These assumptions are useful when the overall goal is to reduce the computational costs. Instead a spectrum is comprised of internal fragments Xu and Ma (2006), complex isotopic distributions and further fragmentation of post-translational modifications themselves Zolg *et al.* (2018). This results in a much more complex picture when all possible associations are considered.

In order to ad hoc learn those complex, long-range fragmentation patterns of high-resolution spectra, we use a deep neural network consisting of dilated convolutions Yu and Koltun (2015). In particular, such a network has a receptive field that spans large portions of the feature-vector or even the entire feature-vector. Similarly, dilated convolutions allow parameter-sharing and hence keep the total number of trainable weights low (compared to a fully connected network). The receptive field grows exponentially with number of layers as shown in Fig. 2. This is facilitated by a dilation rate (size of the gap; steps with zeros between trainable weights of convolutional filter) that increases alongside additional layers. We illustrate how a single output (black arrow) can receive information from any input (black arrows) via a collection of paths (black solid lines) in the block of dilated convolutions. The final dilation rate (table 1.1) matches the input size and facilitates a receptive field that spans the entire feature vector. It should be noted, that the block has nested stacks of convolutional layers and 64 filters per layer, hence the actual model complexity is not fully captured by the simplified illustration here (grey lines in Fig. 2). On the right in Fig. 2 we compare what is drawn versus what was implemented (in parentheses). Additionally, we illustrate how a single output can learn to reflect a specific pair of ions (green path in Fig. 2). Associations between more than two input features can be learned by the model, but not illustrated here. Furthermore, we introduced skip-connections by using convolutions with kernel-size=1 that are added to the stacked convolutions (green inlay on the right in Fig. 2). This allows for inputs to be passed to the output (yellow path in Fig. 2). Additionally, we apply an initial convolution with a kernel-size=1 to the two-vector representation (bottom in Fig. 2). This results in 128 copies that are either identical (or possibly scaled) or select only high intensities (red) or low intensities (blue) or all intensities (blue and read) are passed to the output. That peaks are selected or being kept depends on the sign of the learned weight and bias in combination with the ReLU-activation in the initial layer.

A block of dilated convolutions is commonly called temporal convolutional neural network (TCN) Bai *et al.* (2018). In a TCN, the receptive field grows exponentially and therefore the gradient computation only needs $\log$(distance-between-features) steps. Another promising architecture is a transformer architecture Vaswani *et al.* (2017). A transformer architecture facilitates a gradient computation that is independent of the distance between two features. But the

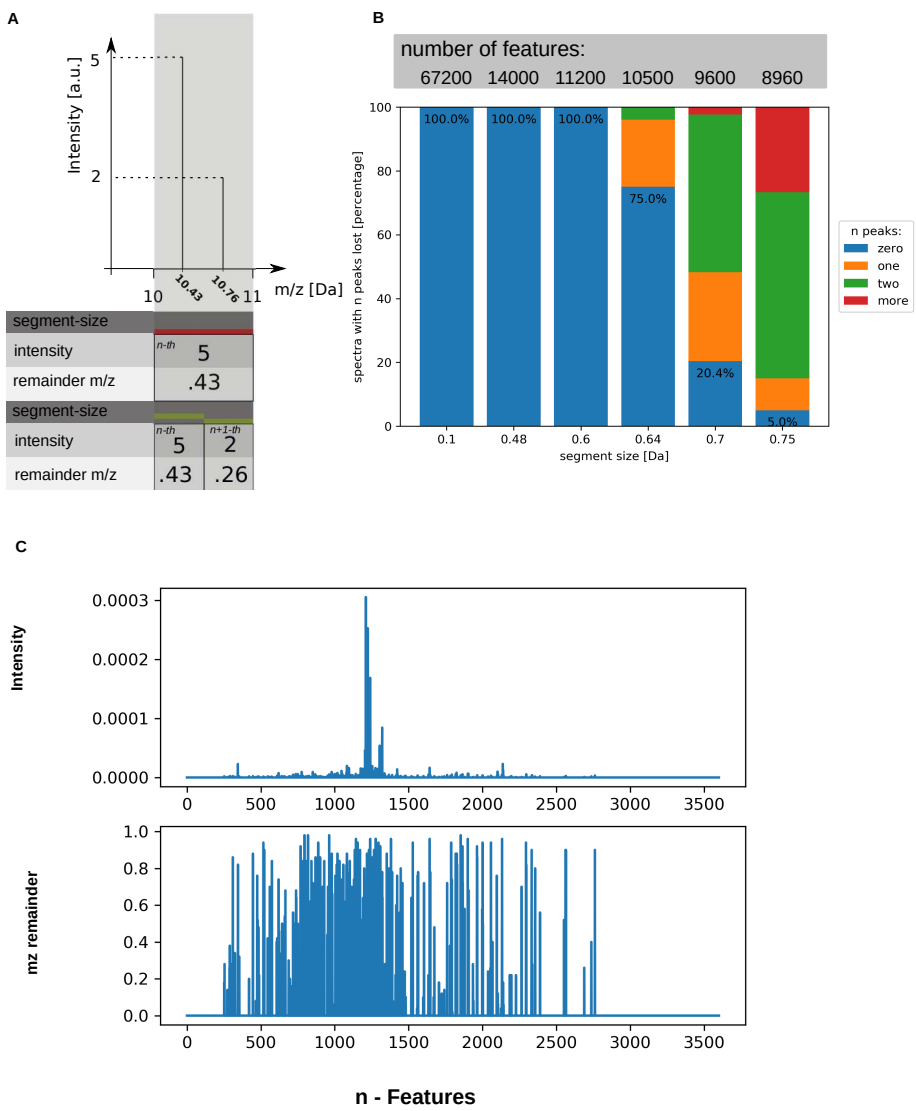

Fig. 1. Two-vector representation, facilitating a loss-less reduction of features. A: Example spectrum of two hypothetical peaks (top). Feature representation as two-vector (bottom): for a large (1 Da) segment size (red) and a smaller (0.5 Da) segment size (green). B: Trade-off between number of features (top, with a windows-size 100-3560 Da) and loss of peaks (y-axis), depending on the chosen segment size (x-axis). C: Tandem mass spectrum represented in two-vectors: intensity (top) and m/z-remainder (bottom). The two feature vectors have 7200 features in total. From the two feature vectors the original peaks-list for this spectrum can be fully recovered (all peaks including their respective m/z values are retained).

computational complexity to perform self-attention (the central operation in a transformer) grows quadratically with input-size as opposed to a TCN, where the complexity due to convolutions scales linearly with the input-size Vaswani *et al.* (2017). Hence, in case of AHLF we preferred a TCN over a transformer. In our TCN, we use padding that conserves the size between the input and the output layer (so-called 'same'-padding). In contrast, TCNs are also usually used in conjunction with 'causal'-padding, e.g. the 'Wavenet' from Google DeepMind is using causal-padding van den Oord *et al.* (2016). This is due to the fact that the latter is modeling audio signals in a time-domain, and feedback from the future to the past is eliminated by 'causal'-padding. In case of a tandem spectrum, fragmentation happens from both ends of a peptide and a spectrum is bi-directional in that sense, hence we decided to stick with 'same'-padding. An output feature is able to receive information from the left and the right part of the previous layer, see Fig. 2. The TCN-block is followed by four dense (fully-connected) layers. The final prediction is facilitated by sigmoid as activation-function, which outputs a score between zero (unphosphorylated) and one (phosphorylated).

A summary of hyperparameters can be found in the supplementary table 1.1. The model has a total number of 6,705,985 trainable model parameters of which 256,832 are trainable parameters resembling the block of dilated convolutions. We used the adaptive learning rate optimizer ADAM Kingma and Ba (2014) with an initial learning rate of $0.5 \cdot 10^{-6}$. A detailed description about the learning process can be found in the supplementary material alongside with a training history of our final model in Fig. 2.3.

2.3 Benchmark data set

For training and testing,we used the combined data set PXD012174 Ochoa *et al.* (2019), containing 112 individual repositories organized according to 101 human cell/tissue types (we count according to the number of zip-files in PXD012174 that contain tandem spectrum raw-files). The individual data sets used phospho-enrichment assays. The combined data set is a

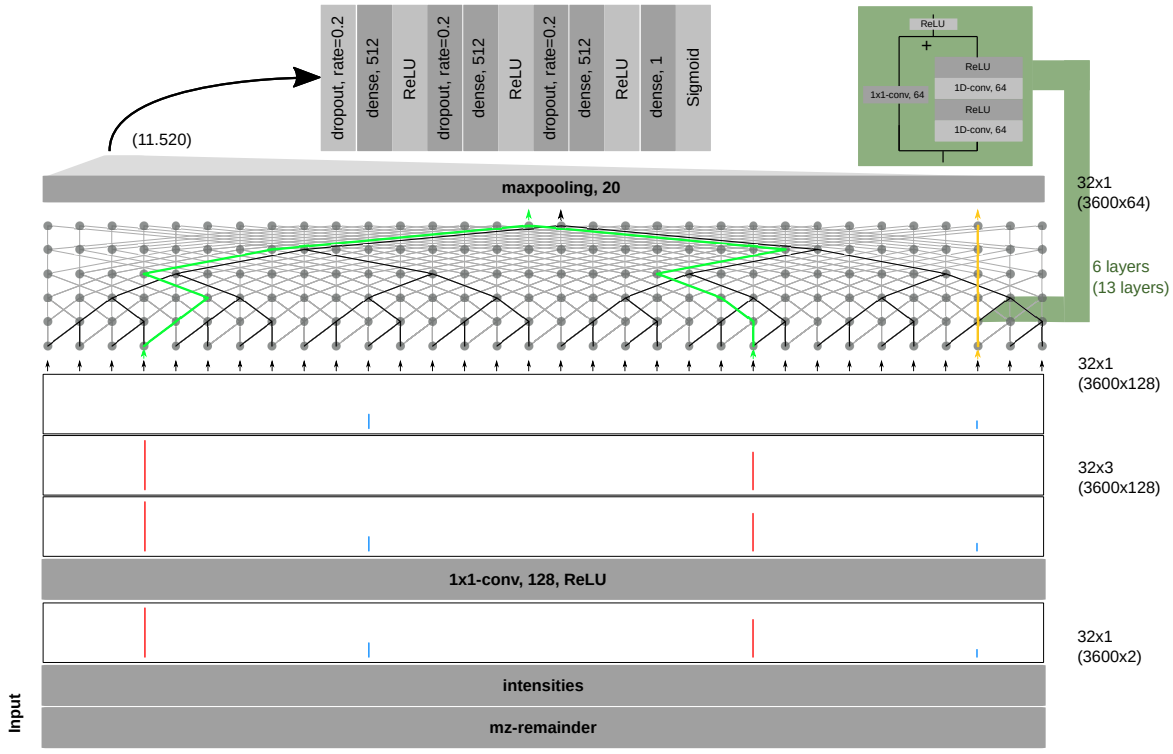

Fig. 2. Illustration of our deep learning model, the hyperparameters are summarized in table 1.1. The receptive field grows exponentially with additional layers due to dilated convolutions as indicated by a possible information-flow (black solid lines): a single output (top) can receive information from any input feature (bottom). Nested stacks of two stacked convolutions and a skip-connection further increase non-linearity (green inlay, right). Only selected parts of the actual network are illustrated here. In parentheses (on the right) the actual tensor size and numbers of filters are compared against what is illustrated (without parentheses) here. We exemplify learnable associations: (i) between a specific pair of ions (green path) or (ii) simply passing inputs through skip-connections (yellow path), other associations are possible to be learned (not illustrated here). An example of a hypothetical intensity vector (bottom) with four peaks of higher intensities (red) and lower intensities (blue) are shown here. The initial convolution with kernel-size=1 creates 128 copies by either selecting higher (middle), lower (top) or all (bottom) intensities.

joint re-analysis with the following error rates: false-discovery rate (FDR) was set to 0.01 at peptide-spectrum match (PSM), protein, and site decoy fraction (PTM site FDR) levels. The minimum score for modified peptides was 40, and the minimum delta score for modified peptides was 6. The combined search results were taken from the 'txt-100PTM'-search results, as described in PXD012174 Ochoa *et al.* (2019).

For each spectrum, a label (phosphorylated or unphosphorylated) was assigned by considering the set of identifications passing the above mentioned score-thresholds. Only if this set exclusively contains either of the two labels, we accepted the respective label and consider the spectrum for further analysis. For example, if the set of identifications for a given spectrum contained both types of peptides (phosphorylated and unphosphorylated peptides) we discarded the spectrum. We argue that this strict voting scheme results in less erroneous labels (compared to e.g. accepting the top-scoring PSM as a label). This ambiguity by a mixed set of PSMs rarely happened. In addition, the usage of the delta score in Maxquant Cox and Mann (2008) (the score difference to the second best identified peptide with a different amino acid sequence) reduces potential conflicting labels. In total, the data set contains 19.2 million PSMs consisting of 10.5 million phosphorylated PSMs and 8.7 million unphosphorylated PSMs. In their original publication, Ochoa et al. reported 11.7 million phosphorylated PSMs Ochoa *et al.* (2019). This discrepancy is caused by our strict assignment of labels.

2.4 Benchmark PhoStar versus AHLF

For the benchmark, we used PhoStar with the provided default parameters (as stated in the documentation and publication of PhoStar), those where the following: m/z resolution was set to 10 ppm, peak-picking depth was set to 10 (per 100 m/z) and score threshold set to 0.5 for all results shown here. PhoStar is closed source and it is not possible to be trained by a user. Therefore, we used the original ensemble model parameters as published by Dorl *et al.* (2018). For 462,464 spectra (2.4% of the benchmark data set) PhoStar was not able predict a classification score (during inference: PhoStar raises an error about mis-matching masses in those cases and does not provide a classification score). To perform a fair comparison we decided to assign a PhoStar-prediction score of 0.5 to those spectra. This equals a random binary-classification. The benchmark data set is comprised of spectra that passed a database search and therefore we assume that no further quality control is needed. Otherwise, allowing the tool PhoStar to exclude spectra before prediction would make a comparison of tools infeasible.

To calculate metrics such as balanced accuracy, F1-Score and the area under the (receiver operating characteristic)-curve (AUC) we used scikit-learn on the predictions from PhoStar and AHLF respectively. Since both tools report a continuous binary-classification score between zero (not being phosphorylated) and one (being phosphorylated). We set the threshold to 0.5 to get the predicted class labels (e.g. in order to calculate accuracy and F1-score). To calculate the AUC we used the predictions scores directly.

3 Results

3.1 Evaluating the performance

To evaluate the performance on the complete set of 19.2 million labeled spectra from PXD012174 we used a four-fold cross validation, see supplementary tables 1.2 and 1.3. This yields four independently trained deep learning models. Hereinafter, we refer to AHLF as the combined results of the four independent models. We compare AHLF against PhoStar (with parameters as explained in the method section). The binary classification results of the 101 individual data sets are shown in Fig. 3. The performance of both tools varies greatly across the different data sets, see Fig. 3. Comparatively, the mean and median are collectively higher for AHLF, see table 1. Furthermore, AHLF shows a smaller variance for each of the three metrics across the individual data sets.

	median AUC	median F1	median Bacc
AHLF	**0.8848**	**0.8332**	**0.7680**
PhoStar	0.8001	0.5838	0.6925
	mean AUC	mean F1	mean Bacc
AHLF	**0.8598**	**0.7425**	**0.7606**
PhoStar	0.8099	0.6132	0.7295
	var AUC	var F1	var Bacc
AHLF	**0.0085**	**0.0488**	**0.0093**
PhoStar	0.0245	0.0833	0.0273

Table 1. Median, mean and variance for AUC, F1-score and balanced accuracy over the individual data sets.

Additionally, we calculated overall metrics on all 19.2 million spectra as if they were a single data set, see table 2. The size (number of spectra) of individual data sets ranges from a few spectra (e.g. ACHN-data set) up to a few millions of spectra for larger data sets (e.g. Jurkat-data set). The overall metrics are predominantly reflecting the performance on larger data sets. Comparing both Tables 1 and 2 suggests that both tools perform better on the larger data sets.

tool	overall AUC	overall F1	overall Bacc
AHLF	**0.9209**	**0.8551**	**0.8368**
PhoStar	0.9168	0.7989	0.8200

Table 2. Overall balanced accuracy, F1-score and AUC.

The individual performance per data set for both tools are shown in the supplementary Tables 1.4 and 1.5 for PhoStar and in supplementary tables 1.6 and 1.7.

We investigated if the performance of both tools is correlated across the different data sets, see supplementary material Fig. 2.1. For example, the Spearman correlation coefficient is 0.864 for the AUC score in supplementary Fig. 2.1A.AHLF and PhoStar are different types of classifiers (neural network versus a random forest classifier) and independently trained. This suggests that some data sets tend to be more challenging for this classification task (i.e. others tend to be more easily classified).

Finally, we studied the performance separately for the used collision methods: collision induced dissociation (CID) and higher energy collision induced dissociation (HCD), see supplementary material Fig. 2.2. For PhoStar, the AUC score for CID spectra resembles a bimodal distribution (supplementary Fig. 2.2A) and the overall AUC of PhoStar is comparatively low, this is mostly due to its weak performance on CID spectra (mean AUC=0.7666), whereas PhoStar performs better on HCD

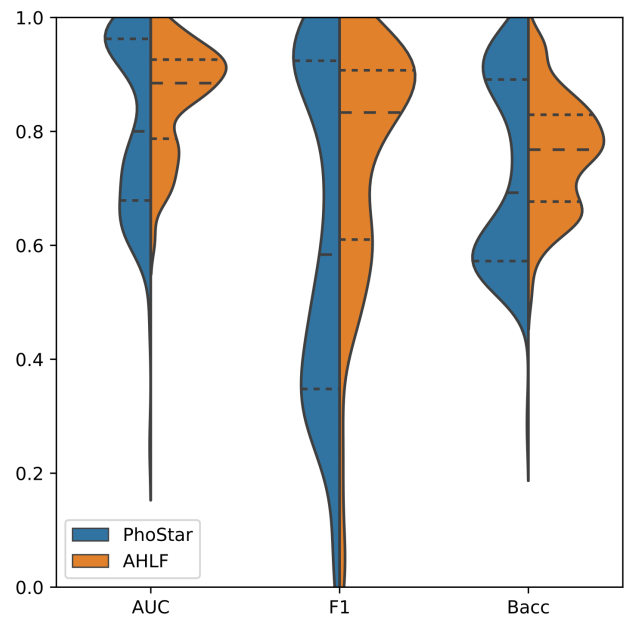

Fig. 3. Violin plot: comparing AHLF (orange) versus PhoStar (blue). AUC, F1-score and balanced accuracy for the individual data sets from PXD012174. Median (large dashed line) and quartiles (small dashed lines) are indicated. Metrics are calculated on holdout data. By combining the individual holdout sets from the cross-validation all 19.2 million data points are used here.

spectra (mean AUC=0.94). Similarly, AHLF has a distinct performance between HCD spectra (mean AUC=0.9078) and CID spectra (mean AUC=0.8381), see (supplementary Fig. 2.2B). For the latter comparison, we calculated the AUC across raw-files. These can be easily separated according to the used collision methods as opposed to the individual 101 data sets that often contain both, HCD- and CID-spectra.

AHLF can compute prediction scores for 1000 preprocessed spectra in 1.085 seconds, on a machine with a Nvidia V100 16GB GPU and an Intel Xeon processor with 14 CPU cores. When spectra are loaded from disk and preprocessed the overall inference time is 10.93 seconds for 1000 spectra. The latter is crucial, in case our framework is used as an integrated pipeline, e.g. before a subsequent database search.

4 Discussion

Our benchmark results suggest a high variance of performance over the different data sets for both tools.Additionally, performances are correlated between both tools, in the sense that some data sets tend to be collectively more challenging than others. This could be due to: erroneous label information resulting from the data base search, biases from larger data sets (e.g. the top-3 largest data sets resemble nearly 40% of the entire combined set), some data sets may be inherently noisier, spectra may have lower resolution or come from less commonly used instruments etc. To further investigate this issue, the entire combined data set could be reorganized in a lab- or even instrument-specific way.

We investigated whether the classification performance depends on the utilized collision method. Both tools tend to perform better on HCD spectra. PhoStar was trained exclusively on HCD spectra. Furthermore, the training set of PhoStar contains spectra from the human HeLa cell line from PXD000612, PXD001333 and PXD001374, the total number of these spectra is 2,007,156 according to Dorl *et al.* (2018). Those HCD spectra are also part of the combined repository PXD012174. Hence,

10.45% of the spectra from PXD012174 were part of the training of PhoStar and thus are part of the benchmark results of PhoStar. This may lead to an overestimation of the performance of PhoStar especially in the supplementary figure 2.2A. Otherwise, we could not re-train PhoStar, which is closed source, leaving us with a potential underestimation of its performance.

Training on simulated data or domain randomization Tobin *et al.* (2017) is a powerful method to improve performance for models and reducing the generalization gap. The prediction of relative intensity profiles is well received in the proteomics community as it improves database search substantially Gabriels *et al.* (2019); Zhou *et al.* (2017); Gessulat *et al.* (2019). This in combination with a sophisticated noise model may become very helpful for domain randomization and training models like AHLF in the near future.

Similarly, traditional preprocessing (e.g. peak-picking) or de-noising may improve the performance of a trained network. But even if preprocessing is well-studied in the context of how it influences database search results Wolski *et al.* (2005, 2006); Renard *et al.* (2009), it remains unclear how those findings translate and effect the performance of deep learning models like AHLF.

Our effort greatly benefits from the vast amount of publicly available mass spectromtry data. In our case, it was particularly helpful that a combined re-analysis of more than one-hundred individual phosphoproteomic data sets was recently published Ochoa *et al.* (2019). This repository makes the label information more accessible than it would be, e.g. from the individual repositories. For potential future work, it is promising that more and more deep learning-specific technical data sets are made available Gessulat *et al.* (2019); Lee *et al.* (2018).

As mentioned in the methods section, we investigate our two-vector representation in a one-to-itself setting. Ideally, one would also investigate such a spectrum representation in a one-to-many setting. For a example, the prediction of AHLF should be invariant to shifts in m/z. Currently, AHLF may waste some of its capacity to handle shifts even though the use of convolutions, maxpooling and dropout are already advantageous against shifts. An investigation of our representation in a one-to-many setting could reveal global conflicts that may be resolved by resampling m/z-values for all spectra and may further improve the performance of AHLF.

Here, we exemplify PTM-calling as an application of our generic deep learning model. But various other applications like dense predictions could be used to predict PTM-surrounding b- and y-ions, in order to highlight the position of the PTM within the spectrum. This is possible because the block of dilated convolutions preserves the input size. The objective function could be adapted accordingly such that the output reflects the PTM-surrounding ions, similar to image segmentation Lin *et al.* (2017).

5 Conclusion

We present a novel method for predicting whether a spectrum originates from a phosphorylated or a non-phosphorylated peptide. Our approach outperforms the current state-of-the-art on this task on a vast repository of data sets from diverse lab environments and experimental settings. With the help of our framework we show that end-to-end training of a generic deep learning model on tandem spectra is feasible. We speculate that this will spark diverse future applications since feature creation is fully integrated in the learning process. To the best of our knowledge, this is the first deep learning model being able to ad hoc learn fragmentation patterns in high-resolution spectra. Additionally, our specific set of hyperparameters may serve as a starting point and speed up hypothesis testing. An implementation with command-line instructions is publicly available from gitlab.com/dacs-hpi/ahlf

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft (grant RE3474/2-2 to BYR) and by the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A537B, 031A533A, 031A538A, 031A533B, 031A535A, 031A537C, 031A534A, 031A532B). We thank Elizabeth Y. Yuu for proofreading this manuscript.

References

Aebersold, R. and Mann, M. (2016). Mass-spectrometric exploration of proteome structure and function. *Nature*, **537**(7620), 347–355.

Bai, S. *et al.* (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.

Bittremieux, W. *et al.* (2018). Fast open modification spectral library searching through approximate nearest neighbor indexing. *bioRxiv*.

Cox, J. and Mann, M. (2008). MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, **26**, 1367.

Cvetesic, N. *et al.* (2016). Proteome-wide measurement of non-canonical bacterial mistranslation by quantitative mass spectrometry of protein modifications. *Scientific Reports*, **6**(1), 28631.

Degnore, J. P. and Qin, J. (1998). Fragmentation of Phosphopeptides in an Ion Trap Mass Spectrometer. **0305**(98).

Devabhaktuni, A. *et al.* (2019). TagGraph reveals vast protein modification landscapes from large tandem mass spectrometry datasets. *Nature Biotechnology*, **37**(4), 469–479.

Dorl, S. *et al.* (2018). PhoStar: Identifying Tandem Mass Spectra of Phosphorylated Peptides before Database Search. *Journal of Proteome Research*, **17**(1), 290–295.

Falick, A. M. *et al.* (1993). Low-mass ions produced from peptides by high-energy collision-induced dissociation in tandem mass spectrometry. *Journal of the American Society for Mass Spectrometry*, **4**(11), 882–893.

Gabriels, R. *et al.* (2019). Updated MS2PIP web server delivers fast and accurate MS2 peak intensity prediction for multiple fragmentation methods, instruments and labeling techniques. *Nucleic Acids Research*, **47**(W1), W295–W299.

Gessulat, S. *et al.* (2019). Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, **16**(6), 509–518.

Goloborodko, A. A. *et al.* (2013). Pyteomics—a Python Framework for Exploratory Data Analysis and Rapid Software Prototyping in Proteomics. *Journal of The American Society for Mass Spectrometry*, **24**(2), 301–304.

Hulstaert, N. *et al.* (2020). ThermoRawFileParser: Modular, Scalable, and Cross-Platform RAW File Conversion. *Journal of Proteome Research*, **19**(1), 537–542.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization.

Kong, A. T. *et al.* (2017). MSFragger: Ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nature Methods*, **14**(5), 513–520.

Kreitzberg, P. A. *et al.* (2019). Alphabet Projection of Spectra. *Journal of Proteome Research*, **18**(9), 3268–3281.

LeCun, Y. *et al.* (2015). Deep learning. *Nature*, **521**(7553), 436–444.

Lee, J.-Y. *et al.* (2018). Proteomics of natural bacterial isolates powered by deep learning-based de novo identification. *bioRxiv*.

Levitsky, L. I. *et al.* (2019). Pyteomics 4.0: Five Years of Development of a Python Proteomics Framework. *Journal of Proteome Research*, **18**(2), 709–714.

Lin, T.-Y. *et al.* (2017). Focal Loss for Dense Object Detection.

Lu, B. *et al.* (2008). Colander: A Probability-Based Support Vector Machine Algorithm for Automatic Screening for CID Spectra of Phosphopeptides Prior to Database Search. *Journal of Proteome Research*, **7**(8), 3628–3634.

May, D. H. *et al.* (2018). A learned embedding for efficient joint analysis of millions of mass spectra. *bioRxiv*.

Nesvizhskii, A. I. *et al.* (2007). Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nature Methods*, **4**(10), 787–797.

Ochoa, D. *et al.* (2019). The functional landscape of the human phosphoproteome. *Nature Biotechnology*.

Potel, C. M. *et al.* (2019). Phosphopeptide Fragmentation and Site Localization by Mass Spectrometry: An Update. *Analytical Chemistry*, **91**(1), 126–141.

Renard, B. Y. *et al.* (2009). When less can yield more - Computational preprocessing of MS/MS spectra for peptide identification. *Proteomics*, **9**(21), 4978–4984.

Strupat, K. *et al.* (2013). High-resolution, accurate-mass orbitrap mass spectrometry–definitions, opportunities, and advantages.

Tobin, J. *et al.* (2017). Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World.

Tran, N. H. *et al.* (2017). De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, **114**(31), 8247–8252.

van den Oord, A. *et al.* (2016). WaveNet: A Generative Model for Raw Audio.

Vaswani, A. *et al.* (2017). Attention Is All You Need.

Vizcaíno, J. A. *et al.* (2017). A community proposal to integrate proteomics activities in ELIXIR. *F1000Research*, **6**, 875.

Wenger, C. D. *et al.* (2010). Sub-part-per-million Precursor and Product Mass Accuracy for High-throughput Proteomics on an Electron Transfer Dissociation-enabled Orbitrap Mass Spectrometer. *Molecular & Cellular Proteomics*, **9**(5), 754–763.

Wolski, W. E. *et al.* (2005). Transformation and other factors of the peptide mass spectrometry pairwise peak-list comparison process. *BMC Bioinformatics*, **6**(1), 285.

Wolski, W. E. *et al.* (2006). Analytical model of peptide mass cluster centres with applications. *Proteome science*, **4**, 18.

Xu, C. and Ma, B. (2006). Complexity and scoring function of MS/MS peptide de novo sequencing. *Computational systems bioinformatics. Computational Systems Bioinformatics Conference*, pages 361–9.

Xu, L. *et al.* (2020). Machine Learning in Mass Spectrometric Analysis of DIA Data. *PROTEOMICS*, page 1900352.

Yu, F. and Koltun, V. (2015). Multi-Scale Context Aggregation by Dilated Convolutions. *preprint*.

Zhou, X.-X. *et al.* (2017). pDeep: Predicting MS/MS Spectra of Peptides with Deep Learning. *Analytical Chemistry*, **89**(23), 12690–12697.

Zolg, D. P. *et al.* (2018). ProteomeTools: Systematic Characterization of 21 Post-translational Protein Modifications by Liquid Chromatography Tandem Mass Spectrometry (LC-MS/MS) Using Synthetic Peptides. *Molecular & Cellular Proteomics*, **17**(9), 1850 LP – 1863.